A Randomized Algorithm for Gossiping in Radio Networks

Marek Chrobak* Leszek Gąsieniec† Wojciech Rytter ‡

Abstract

We present an $O(n \log^4 n)$-time randomized algorithm for gossiping in radio networks with unknown topology. This is the first algorithm for gossiping in this model whose running time is only a poly-logarithmic factor away from the optimum. The fastest previously known (deterministic) algorithm for this problem works in time $O(n^{3/2} \log^2 n)$.

1 Introduction

The two classical problems of disseminating information in computer networks are *broadcasting* and *gossiping*. In broadcasting, we want to distribute a given message from a distinguished source node to all other nodes in the network. In gossiping, each node v in the network initially contains a message m_v, and we wish to distribute each message m_v to all nodes in the network. In both problems, we would like to minimize the time needed to complete the task.

In radio networks, a message transmitted by a processor is sent to all processors within its range. The range relation is represented by a graph of nodes with directed edges between them. All processors work synchronously, and if a processor u transmits a message m at time step t, the message reaches each neighbor v of u at the same time step. Node v will successfully receive m only if u is the only processor, among those whose range contains v, that transmits at time t. Since the communication links are uni-directional, there is no feedback mechanism in the network (see, for example, [20]), and thus, in general, a node does not know for certain whether its transmissions were successful. Further, we assume that collisions cannot be detected, that

*Department of Computer Science, University of California, Riverside, CA 92521. Email: marek@cs.ucr.edu.
Research done while the author was visiting University of Liverpool. Research supported by grant EPSRC GR/N09077 and by NSF grant CCR-9988360.

†Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK. Email: leszek@csc.liv.ac.uk. Research supported by grant EPSRC GR/N09077.

‡Instytut Informatyki, Uniwersytet Warszawski, Banacha 2, 02–097, Warszawa, Poland, and Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK. Email: rytter@csc.liv.ac.uk. Research supported by grant EPSRC GR/N09077.

is, if messages from two or more processors reach v at time t, v does not know that the collision occurred. This is motivated by situations when message collisions are difficult to distinguish from background noise, in which protocols that do not depend on the accuracy of the collision detection mechanism (see [13, 12]) will be more reliable.

We focus on gossiping algorithms that do not use any information about network topology. Such topology-independent algorithms are useful in networks with mobile users or unstable topologies, since then one does not need to change or reconfigure the protocol after topology changes. As long as the network is strongly connected and no changes occur during the actual execution of the algorithm, the task of gossiping will complete successfully. The strong connectivity assumption is necessary for gossiping to be meaningful.

Past work. Most of the previous work on radio networks focussed on broadcasting. If the topology of the network is known to all processors, Gaber and Mansour [14] showed that broadcasting can be achieved in time $O(D + \log^5 n)$, where D is the network diameter. Diks *et al* [10] gave efficient broadcasting algorithms for special types of known networks. It is also known that computing an optimal broadcast schedule for a given network is NP-hard, even for points in the plane, where the graph is induced by node ranges, see [8, 26].

For networks with unknown topology, Bar-Yehuda *et al* [3] gave a randomized algorithm that achieves broadcast in expected time $O(D \log n + \log^2 n)$. This is very close to the lower bound of $\Omega(D \log(n/D))$, by Kushilevitz and Mansour [18], and it matches this lower bound for a wide range of depth values, for example when $D = \Theta(n^{1-\epsilon})$, for any $\epsilon > 0$. Further, if D is a constant, it also matches the lower bound of $\Omega(\log^2 n)$ for constant diameter networks, obtained by Alon *et al* [1].

In the deterministic case, Bar-Yehuda *et al* [3] gave an $\Omega(n)$ lower bound for constant diameter networks. For general networks, the best currently known lower bound of $\Omega(n \log n)$ was obtained by Bruschi and del Pinto [5], and, independently by Chlebus *et al* [6]. In [6], the authors also present a broadcast algorithm with time complexity $O(n^{11/6})$ – the first sub-quadratic upper bound. This upper bound was later improved to $O(n^{5/3} \log^3 n)$ by De Marco and Pelc [11]. Chlebus *et al* [7] developed several broadcasting algorithms, including one with time complexity $O(n^{3/2})$. In an independent work, using a probabilistic construction, Peleg [23] gave an $O(n^{3/2}\sqrt{\log n})$ upper bound. Recently, Chrobak, Gąsieniec and Rytter [9] presented a deterministic algorithm for broadcasting with time complexity $O(n \log^2 n)$, thus nearly matching the lower bound of $\Omega(n \log n)$ from [5, 6]. As in [23], this broadcasting algorithm was constructed using a probabilistic argument.

The problem of gossiping has been intensely studied in various network models (see, for example, [15]). For radio networks, Ravishankar and Singh [24, 25] studied gossiping algorithms for some restricted topologies, including paths and rings, under probabilistic assumptions on the spatial distribution of nodes. In our previous work, [9], we developed a deterministic algorithm for gossiping with time complexity $O(n^{3/2}\log^2 n)$, which, to our knowledge, is the only sub-quadratic algorithm for gossiping in radio networks with unknown topology.

Our results. In this paper we give a randomized $O(n\log^4 n)$-time algorithm for gossiping in radio networks with unknown topology. Our basic algorithm is Monte Carlo and it has the following performance characteristics: for any $0 < \epsilon < 1$, in time $O(n\log^3 n\log(n/\epsilon))$ it achieves broadcast with probability at least $1-\epsilon$. This easily yields a Las Vegas algorithm with expected running time $O(n\log^4 n)$.

2 Preliminaries

Radio networks. A *radio network* (see [3, 7]) is defined as an n-node directed graph whose nodes are assigned unique identifiers from the set $\{1, 2, \ldots, n\}$. Throughout the paper, for gossiping to be meaningful, we assume that the network is strongly connected. If there is an edge from u to v, then we say that v is an *out-neighbor* of u and u is an *in-neighbor* of v.

Initially, each node v contains a message m_v and has no other information. The time is divided into discrete time steps. All nodes start simultaneously, have access to a common clock, and work synchronously. (As noted by Peleg [23], the assumption about a common clock is not necessary.) At any time step, a node can be in one of two states: the *receiving state* or the *transmitting state*. A gossiping algorithm is a protocol that for each identifier *id* and for each time step t, given all past messages received by *id*, specifies the state of *id* at time t. If *id* transmits at time t, the protocol specifies the message. A message m transmitted at time t from a node u is sent instantly to all its out-neighbors. However, an out-neighbor v of u receives m at time step t only if v is in the receiving state and if no collision occurred, that is, if the other in-neighbors of v do not transmit at time t at all. Further, collisions cannot be distinguished from background noise. If v does not receive any message at time t, it knows that either none of its in-neighbors transmitted at time t, or that at least two did, but it does not know which of these two events occurred.

The running time of a gossiping algorithm is the smallest t such that for any strongly connected network topology, and for any assignment of identifiers to the nodes, each node receives all messages m_v no later than at step t.

Simplifying assumptions. For clarity of presentation, we will present our algorithms as if the nodes knew n, the size of the network. This assumption can be eliminated by a standard doubling technique (see [6, 7]) that works as follows: We organize the computation into phases, and we modify a given algorithm so that in phase i only nodes with labels at most 2^i participate in the algorithm. This does not change the asymptotic running time.

Further, we will also assume throughout the paper that n is a power of 2. For other n, the processors can execute the algorithm for the nearest power of 2 larger than n, without changing the asymptotic running time.

Notation. By V we denote the set of nodes, and individual nodes are denoted by letters $u, v, \ldots$. Messages are denoted by letter m, possibly with indices. The message originating from a node v is denoted by m_v. The whole set of initial messages is $M = \{m_v : v \in V\}$. During the computation, each node v will store a set of messages M_v that have been received by v so far. Initially, $M_v = \{m_v\}$. Without loss of generality, whenever a node is in the transmit mode, we can assume that it transmits the whole M_v. This is achieved by a procedure denoted *transmit*(M_v). Procedure *receive*() puts v in the receive mode and it returns the received message, or *null* if no message has been received.

3 Limited Broadcast

One component of our algorithm is a procedure for *limited broadcasting.* Given an integer k and a node v, the goal of limited broadcasting is to send the message M_v to at least k nodes in the network. We refer to v as the *source node* or the node that *initiates* the broadcast, and to M_v as the *source message.*

In [9], the broadcasting algorithm is defined by a sequence $\bar{S} = S_0 S_1 \ldots$ of subsets of $\{1, 2, \ldots, n\}$. At time t, any node v that has already received the source message checks whether $v \in S_t$. If so, v transmits the message, otherwise v is quiet. We modify the algorithm from [9], so that it performs the broadcasting procedure for only $O(k \log^2 n)$ steps. Below, we appropriately refine the correctness proof, since the proof from [9] is not sufficient for our purpose.

The pseudo-code for the algorithm executed by each node is given below. Each node v has a boolean flag *active*$_v$ that indicates whether v is *active*, that is, whether v has received a source message. Each iteration of the for-loop lasts one time step. The value of constant γ will be determined later.

Procedure $\text{LtdBroadcast}_v(k)$.
for $\tau = 0, 1, \ldots, \gamma k \log^2 n - 1$ **do**
  if $active_v$ and $v \in S_\tau$ **then** $transmit(M_v)$
  else
    $m \leftarrow receive()$;
    if $m \neq null$ **then**
      $M_v \leftarrow M_v \cup \{m\}$;
      $active_v \leftarrow true$

Lemma 1 *Assume that initially exactly one node u is active and that all nodes v begin executing* $\text{LtdBroadcast}_v(k)$ *simultaneously. Then, for some constant γ independent of k, after the computation is complete at least k nodes will receive the message from u.*

Proof: Without loss of generality, assume $n \geq 2$. In [9], it was shown that there is a constant α such that for each $j = 0, \ldots, \log n$ there is a family $\bar{S}_j = (S_{j,0}, S_{j,1}, \ldots, S_{j,a_j-1})$ of $a_j = \alpha 2^j \log n$ sets with the following property:

(*) For any two disjoint sets X, Y with $2^{j-1} \leq |X| \leq 2^j$ and $|Y| \leq 2^j$ there exists a set $S_{j,i}$ in $\bar{S}_j$ such that $|X \cap S_{j,i}| = 1$ and $Y \cap S_{j,i} = \emptyset$.

Let $\gamma = 12\alpha + 2$. The sequence $\bar{S}$ consists of *stages*, with each stage, except possibly the last one, having $\log n + 1$ steps. Note that the number of stages is $\lceil \gamma k \log^2 n / (\log n + 1) \rceil \geq 6\alpha k \log n + 1$. The transmission set at the jth step of stage s, that is S_τ for $\tau = s(\log n + 1) + j$, is $S_{j, s \bmod a_j}$.

Among the active nodes we distinguish two types of nodes: *frontier nodes*, which still have inactive out-neighbors, and *inner nodes*, which don't. If there is a time step $t < \gamma k \log^2 n$ when there are k or more frontier nodes, then the lemma holds trivially. So from now on we assume that at each step the number of frontier nodes is less than k.

We define a sequence of stages $s_0 = 0, s_1, \ldots, s_{l+1} = 6\alpha k \log n$, where $1 \leq s_{c+1} - s_c \leq 2\alpha k \log n$ for all c. Denote by i_c and f_c the number of inner and frontier nodes when stage s_c is about to start. We will choose $s_1, \ldots, s_l$ so that the following invariant holds for each $c \leq l$:

$$2i_c + f_c \quad \geq \quad \frac{s_c}{2\alpha \log n}. \tag{1}$$

Given (1), we can prove the lemma as follows. The number of nodes that have received the message when stage s_{l+1} ends is at least

$$\begin{aligned} i_l + f_l &\geq \tfrac{1}{2}(2i_l + f_l) \\ &\geq \frac{s_l}{4\alpha \log n} \\ &\geq \frac{s_{l+1} - 2\alpha k \log n}{4\alpha \log n} \\ &= k. \end{aligned}$$

So it is sufficient to construct $s_1, \ldots, s_l$ that satisfy (1). We define these stages inductively. For $c = 0$ we have $s_0 = 0$, $i_0 = 0$ and $f_0 = 1$, so (1) holds. Suppose we have determined some s_c. If $s_c > 4\alpha k \log n$, set $l = c$ and we are done. Otherwise, we proceed as follows.

Let F be the set of frontier nodes at the beginning of stage s_c, and let g be such that $2^{g-1} \leq |F| < 2^g$. For each $j = 1, \ldots, g$, let Y_j be the set of nodes that received the message in stages $s_c, s_c + 1, \ldots, s_c + a_j - 1$ (but were inactive when stage s_c started).

We have two sub-cases:

<u>Case 1</u>: There is j for which $|Y_j| \geq 2^j$. In this case, take $s_{c+1} = s_c + a_j$. At least $|Y_j|$ new nodes received the message, so

$$\begin{aligned} 2i_{c+1} + f_{c+1} &\geq 2i_c + f_c + |Y_j| \\ &\geq \frac{s_c}{2\alpha \log n} + 2^j \\ &\geq \frac{s_{c+1}}{2\alpha \log n}. \end{aligned}$$

<u>Case 2</u>: For each j we have $|Y_j| \leq 2^j$. We show that in this case all nodes in F will become inner after a_g stages.

Fix any node v that is inactive when stage s_c starts, and whose set X of in-neighbors in F is not empty. Pick j such that $2^{j-1} \leq |X| < 2^j$. Since $|Y_j| \leq 2^j$, by property $(*)$, family $\bar{S}_j$ contains a set $S_{j,i}$ that hits X and avoids Y_j. This $S_{j,i}$ will occur in one of the stages $s_c, s_c + 1, \ldots, s_c + a_j - 1$.

All in-neighbors of v are either in X or are inactive when stage s_c starts. When we use $S_{j,i}$ for transmission then:

(i) exactly one in-neighbor of v in X will transmit because $|S_{j,i} \cap X| = 1$,

(ii) the nodes from Y_j will not interfere because $S_{j,i} \cap Y_j = \emptyset$,

(iii) the nodes that were inactive at the beginning of stage s_c and are not in Y_j remain inactive when $S_{j,i}$ is issued, so they will not transmit.

Therefore v will receive the message when $S_{j,i}$ is issued (unless it has already received it earlier). Since v was an arbitrary inactive out-neighbor of F, we conclude that all nodes in F will become inner after a_g stages.

Take $s_{c+1} = s_c + a_g$. In this case $i_{c+1} \geq i_c + f_c$ and $f_c \geq 2^{g-1}$, so

$$\begin{aligned} 2i_{c+1} + f_{c+1} &\geq 2i_c + f_c + f_c \\ &\geq \frac{s_c}{2\alpha \log n} + 2^{g-1} \\ &\geq \frac{s_{c+1}}{2\alpha \log n}. \end{aligned}$$

We thus proved that (1) holds for s_{c+1}. Further, since $|F| \leq k$, we have $a_g = \alpha 2^g \log n \leq 2\alpha k \log n$, and thus in both sub-cases we have $s_{c+1} - s_c \leq 2\alpha k \log n$. Thus the proof of the lemma is now complete. □

4 Distributed Coupon Collection

In each phase of our algorithm we will attempt to distribute each message m_v to some number of nodes, by performing a sequence of limited broadcasts. We need to achieve two goals. To obtain a good running time, the number of limited broadcasts must be small. Further, each message m_v should participate in at least one limited broadcast, that is, m_v must be in at least one M_u, for some u that initiates a limited broadcast.

To choose the nodes v for which we initiate a limited broadcast, we use randomization. The principle behind the random process we use is similar to that in the coupon collector's problem. There are two differences though. First, each coupon may have several copies. Second, since we do not have enough time to coordinate the choices, we cannot guarantee that exactly one node will initiate broadcasting.

We think of V as a set of n bins and M as a set of n coupons. Each coupon has at least k copies, each copy belonging to a different bin. M_v is the set of coupons in bin v. Consider the following process: At each step, we open bins at random, by choosing each bin, independently, with probability $1/n$. If exactly one bin, say v, is opened, all coupons from M_v are *collected.* If no bin is opened, or if two or more bins are opened, a failure occurs and no coupons are collected. How many steps do we need so that with high probability (a copy of) each coupon is collected?

The distributed coupon collection can be written in pseudocode as follows.

Procedure *DistCouponColl*(s).
  repeat s **times**
    for each bin v **do**
      with probability $1/n$ **do** open v
        else close v
    if exactly one bin v is opened **then**
      collect all coupons from M_v

Lemma 2 *Assume we have n bins and n coupons, and that each coupon has at least k copies, each copy belonging to a different bin. Let δ be a given constant, $0 < \delta < 1$, and $s = (4n/k)\ln(n/\delta)$. Then, after performing DistCouponColl(s), with probability at least $1 - \delta$, all coupons will be collected.*

Proof: The lemma is trivially true for $n = 1$, so we can assume that $n \geq 2$. Let $\mathcal{X}_{m,j}$ be the event that coupon m is collected at a given step j. Then $\Pr[\mathcal{X}_{m,j}]$ is the probability that one bin containing m is opened and all other bins are closed.

For all m and j we have

$$\begin{aligned} \Pr[\mathcal{X}_{m,j}] &\geq \frac{k}{n}\left(1 - \frac{1}{n}\right)^{n-1} \\ &\geq \frac{k}{n}\left(1 - \frac{1}{n}\right)^{n} \\ &\geq \frac{k}{4n}, \end{aligned}$$

where the last inequality follows from the fact that the sequence $(1 - 1/n)^n$ is monotonely increasing. The probability that some m is not collected in s steps is

$$\begin{aligned} \Pr\left[\bigvee_{m\in M} \bigwedge_{j\leq s} \neg\mathcal{X}_{m,j}\right] &\leq n\left(1 - \frac{k}{4n}\right)^{s} \\ &\leq ne^{-sk/4n} \\ &\leq \delta, \end{aligned}$$

by the definition of s. $\square$

5 The Gossiping Algorithm

We now present our Monte Carlo algorithm for gossiping. Each node v performs its version of the algorithm.

Algorithm $\textsc{RandGossip}_v(\epsilon)$.

$\delta \leftarrow \epsilon/\log n$

for $i = 0, 1, \ldots, \log n - 1$ **do**

$Phase_i$:

$s_i \leftarrow (4n/2^i)\ln(n/\delta)$

repeat s_i **times**

for each node v **do** (in parallel)

with probability $1/n$ **do** $active_v \leftarrow true$

else $active_v \leftarrow false$

$\textsc{LtdBroadcast}_v(2^{i+1})$

Theorem 1 *Let ϵ, $0 < \epsilon < 1$, be a given constant. With probability at least $1 - \epsilon$, Algorithm* $\textsc{RandGossip}(\epsilon)$ *completes gossiping in time $O(n \log^3 n \log(n/\epsilon))$.*

Proof: In phase i, the call to $\textsc{LtdBroadcast}_v(2^{i+1})$ costs $O(2^i \log^2 n)$, so phase i costs $O(s_i 2^i \log^2 n) = O(n \log^2 n \log(n/\delta))$. Since we have $\log n$ phases and $\delta = \epsilon/\log n$, this implies the bound on the running time of the algorithm.

Initially, when phase 0 starts, each m_u is in one set M_v, namely in M_u. The algorithm attempts to maintain the invariant that after phase i each m_u is in at least 2^{i+1} sets M_v. If this invariant is preserved at each phase, the gossiping will complete successfully, since after phase $\log n - 1$ each m_u will be in n sets M_v. Thus it is sufficient to prove that the probability that the invariant fails in some phase is at most ϵ.

The process of distributing messages in a given phase i is equivalent to the distributed coupon collection problem described in the previous section, where we view each node as a bin, and active nodes correspond to open bins. Thus, by Lemma 2, the probability that the invariant fails in this phase, assuming that it has not failed in any previous phase, is at most δ. So, overall, the probability of failure in some phase is at most $\log n \cdot \delta = \epsilon$. $\Box$

To obtain a Las Vegas algorithm, run $\textsc{RandGossip}(\epsilon)$ with $\epsilon = 1/n$. After the algorithm halts, each node that has not received n different initial messages, announces that a failure occurred. Since all these nodes send out the same message, this can be achieved with just one broadcast. If B is the running time of the broadcasting algorithm, after B steps each node knows whether the gossiping was successful or not. If the gossiping failed, we run a naive, deterministic $\textsc{RoundRobin}$ algorithm that consists of n rounds, with each node transmitting once in each round. This will achieve gossiping in time $O(n^2)$. Overall, the expected running time will be $O((1 - 1/n)n \log^4 n + (1/n)n^2) = O(n \log^4 n)$. Concluding, we get the following theorem.

Theorem 2 *There exists a randomized Las Vegas algorithm for gossiping with running time $O(n \log^4 n)$.*

6 Final Comments

Several open problems remain. The only known lower bounds for gossiping are those for broadcasting [1, 5, 7], a seemingly easier problem. The gap between lower and upper bounds is particularly wide in the deterministic case: between $\Omega(n \log n)$ and $O(n^{3/2} \log^2 n)$. Closing or at least reducing this gap is an interesting open problem.

Our algorithm is probably not optimal. One possible research direction is to investigate whether one can improve the running time of our algorithm by using the randomized broadcasting algorithm from [3] to perform limited broadcast. It is not quite clear whether the algorithm from [3] can be modified to satisfy Lemma 1. Further, the analysis of this modified algorithm will probably be much more complicated.

Improving the bounds on gossiping or broadcasting may resolve the question whether gossiping is harder than broadcasting in the radio network model (both in the randomized and the deterministic case). Our result implies that, at least in the randomized case, the difference is at most a poly-logarithmic factor.

References

[1] N. Alon, A. Bar-Noy, N. Linial and D. Peleg, A lower bound for radio broadcast, *Journal of Computer and System Sciences* 43 (1991) 290–298.

[2] N. Alon, A. Bar-Noy, N. Linial and D. Peleg, Single round simulation of radio networks, *Journal of Algorithms* 13 (1992) 188–210.

[3] R. Bar-Yehuda, O. Goldreich, and A. Itai, On the time complexity of broadcast in radio networks: An exponential gap between determinism and randomization, *Journal of Computer and System Sciences* 45 (1992) 104–126.

[4] R. Bar-Yehuda, A. Israeli, and A. Itai, Multiple communication in multi-hop radio networks, *SIAM Journal on Computing* 22 (1993) 875–887.

[5] D. Bruschi and M. del Pinto, Lower bounds for the broadcast problem in mobile radio networks, *Distributed Computing* 10 (1997) pp. 129-135

[6] B.S. Chlebus, L. Gąsieniec, A.M. Gibbons, A. Pelc, and W. Rytter, Deterministic broadcasting in unknown radio networks, in *Proc. 11th Ann. ACM-SIAM Symp. on Discrete Algorithms*, San Francisco, California, 2000, pp. 861–870.

[7] M. Chlebus, L. Gąsieniec, A. Östlin and J.M. Robson, Deterministic broadcasting in radio networks, *Proc. 27th International Colloquium on Automata, Languages and Programming,* ICALP'2000, to appear.

[8] I. Chlamtac and S. Kutten, On broadcasting in radio networks - problem analysis and protocol design, *IEEE Transactions on Communications* 33 (1985) 1240–1246.

[9] M. Chrobak, L. Gąsieniec and W. Rytter, Fast broadcasting and gossiping in radio networks. To be presented at *FOCS 2000.*

[10] K.Diks, E.Kranakis, D.Krizanc, A.Pelc, The impact of knowledge on broadcasting time in radio networks, In *Proc. 7th European Symp. on Algorithms* (1999) 41-52

[11] G. De Marco and A. Pelc, Faster broadcasting in unknown radio networks, unpublished manuscript, 1999.

[12] A. Ephremides and B. Hajek, Information theory and communication networks: an unconsummated union, *IEEE Trans. on Inf. Theory*, IT-44 (1998) 2416-2434.

[13] R. Gallager, A perspective on multiaccess communications, *IEEE Trans. on Inf. Theory*, IT-31 (1985) 124-142.

[14] I. Gaber and Y. Mansour, Broadcast in radio networks, in *Proc. 6th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 1995, pp. 577–585.

[15] S.T. Hedetniemi, S.T. Hedetniemi, and A.L. Liestman, A survey of gossiping and broadcasting in communicatio nnetworks, *Networks* 18 (1988) 319–349.

[16] J. Komlos and A.G. Greenberg, An asymptotically nonadaptive algorithm for conflict resolution in multiple-access channels, *IEEE Trans. on Inf. Theory*, IT-31 (1985) 302-306.

[17] E. Kranakis, D. Krizanc and A. Pelc, Fault-tolerant broadcasting in radio networks, in *Proc. 6th European Symposium on Algorithms*, Venice, Italy, 1998, Springer LNCS 1461, pp. 283–294.

[18] E. Kushilevitz and Y. Mansour, An $\Omega(D \lg(N/D))$ lower bound for broadcast in radio networks, *SIAM Journal on Computing* 27 (1998) 702–712.

[19] E. Kushilevitz and Y. Mansour, Computation in noisy radio networks, in *Proc. 9th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 1998, pp. 236–243.

[20] J.L. Massey and P. Mathys, The collision channel without feedback, *IEEE Trans. on Inf. Theory*, IT-31 (1985) 192-204.

[21] R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.

[22] K. Pahlavan and A. Levesque, *Wireless Information Networks*, Wiley-Interscience, New York, 1995.

[23] D. Peleg, Deterministic radio broadcast with no topological knowledge, manuscript, 2000.

[24] K. Ravishankar and S. Singh, Asymptotically optimal gossiping in radio networks, *Discr. Appl. Math.* 61 (1995) 61–82.

[25] K. Ravishankar and S. Singh, Gossiping on a ring with radios, *Parallel Processing Letters* 6 (1996), p. 115-126.

[26] A. Sen and M. L. Huson, A new model for scheduling packet radio networks, in *Proc.* 15*th Ann. Joint Conference of the IEEE Computer and Communication Societies*, 1996, pp. 1116–1124.